

**Amendments to the Claims**

Please amend Claims 4, 5 and 24. The Claim Listing below will replace all prior versions of the claims in the application:

**Claim Listing**

1-3. (Canceled)

4. (Currently Amended) A method for updating a multi-level lookup table comprising the steps of:

providing a default route memory for storing an inherit indicator to indicate that a default route corresponding to the route associated with the root of a subtree is inherited from another subtree, wherein the inherited default route is forwarded by a default index pipeline;

sharing the default route by nodes in the subtree; and

modifying the default route of the root of the subtree, in a single write, by updating the default route memory of the root of the subtree to include a route index value by performing a single write to the default route memory.

5. (Currently Amended) A method for updating a multi-level lookup table comprising the steps of:

providing a default route memory for storing a default route for a subtree, wherein the subtree is a dense subtree and a dense subtree descriptor associated with the subtree includes the default route memory;

sharing the default route by nodes in the subtree; and

modifying the default route of a root node, in a single write, by updating the default route memory of the root node to include a route index value by performing a single write to the default route memory.

6. (Original) The method as claimed in Claim 5 wherein the default route is shared by storing a use default indicator in a mapper entry associated with at least one node in the subtree.
7. (Original) The method as claimed in Claim 6 further comprising the step of:  
returning the default route as a result of a search of the lookup table upon detecting the use default indicator stored in the mapper entry.
8. (Previously Presented) The method as claimed in Claim 4 wherein the subtree is a sparse subtree, the number of routes in the sparse subtree is greater than one and a sparse subtree entry associated with the subtree includes the default route memory.
9. (Previously Presented) The method as claimed in Claim 4 wherein the subtree is a sparse subtree, the number of routes in the sparse subtree is one, and a default route memory associated with the sparse subtree stores the default route.
- 10-12 (Canceled)
13. (Previously Presented) A multi-level lookup table comprising:  
a default route memory which stores an inherent indicator to indicate that a default route associated with the root of the subtree is inherited from another subtree;  
default logic which returns the default route as a result of a search of the lookup table;  
and  
a default index pipeline which forwards the inherited default route.
14. (Previously Presented) A multi-level lookup table comprising:  
a default route memory which stores a default route shared by nodes in a subtree,  
wherein the subtree is a dense subtree and a dense subtree entry associated with the subtree includes the default route memory; and

default logic which returns the default route as a result of a search of the lookup table.

15. (Original) The multi-level lookup table as claimed in Claim 14 further comprising:  
a mapper entry associated with at least one node in the subtree, the mapper entry stores a use default indicator which indicates that the default route stored in the default route memory is the default route for the at least one node, the default route is modified by performing a single write to the default route memory.
16. (Original) The multi-level lookup table as claimed in Claim 15 wherein the default logic returns the default route upon detecting the use default indicator stored in the mapper entry.
17. (Previously Presented) The multi-level lookup table as claimed in Claim 13 wherein the subtree is a sparse subtree, the number of routes in the sparse subtree is greater than one and a sparse subtree entry associated with the subtree includes the default route memory.
18. (Previously Presented) The multi-level lookup table as claimed in Claim 13 wherein the subtree is a sparse subtree, the number of routes in the sparse subtree is one, and the default route memory is stored in a default mapper entry associated with the sparse subtree descriptor.
- 19-21 (Canceled)
22. (Previously Presented) A multi-level lookup table comprising:  
a default route memory which stores an inherit indicator to indicate that a default route associated with the root of the subtree is inherited from another subtree;  
means for returning the default route as a result of a search of the lookup table; and  
means for forwarding the inherited default index.

23. (Previously Presented) A multi-level lookup table comprising:
- a default route memory which stores a default route for nodes in a subtree, wherein the subtree is a dense subtree and a dense subtree descriptor associated with the subtree includes the default route memory; and
  - means for returning the default route as a result of a search of the lookup table upon detecting the use default indicator stored in the mapper entry.
24. (Currently Amended) A multi-level lookup table comprising:
- a default route memory which stores a default route for nodes in a subtree;
  - means for returning the default route as a result of a search of the lookup table upon detecting the use default indicator stored in the mapper entry; and
  - means for sharing the default route amongst nodes in the subtree by storing a use default indicator in a mapper entry associated with at least one node in the subtree and modifying the default route of a root node, in a single write, by updating the default route memory of the root node to include a route index value by performing a single write to the default route memory.
25. (Previously Presented) The lookup table as claimed in Claim 22 wherein the subtree is a sparse subtree the number of routes in the sparse subtree is greater than one and a sparse subtree descriptor associated with the sparse subtree includes the default route memory.
26. (Previously Presented) The lookup table as claimed in Claim 22 wherein the subtree is a sparse subtree, the number of routes in the sparse subtree is one, and the default route memory is stored in a default mapper entry associated with the sparse subtree descriptor.